

P2PSIP Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 15, 2008

L. Li
Ch. Zhang
Y. Wang
Y. Ji
MINE/BUPT
November 12, 2007

A SIP-based P2PSIP Client Protocol
draft-li-p2psip-client-protocol-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on May 15, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document presents the motivation, design guidelines and high level descriptions of peer-to-peer session initiation protocol (P2PSIP) client protocol, which is spoken between P2PSIP peers and P2PSIP clients. In this document, the necessity of P2PSIP Client is identified firstly. Then, two fundamental guidelines for our design are explained in details. Following these guidelines, the design of

proposed protocol and new extensions to existing SIP (session initiation protocol) are described. Such a SIP-based protocol not only makes itself backward-compatible with the vast existing SIP devices, but also provides extra functions to satisfy some particular requirements raised by P2PSIP.

Table of Contents

1.	Introduction	4
1.1.	Requirements Language	5
2.	Concepts and Terminologies	5
3.	The necessity of P2PSIP Client	6
4.	Our Design Guidelines	7
4.1.	Why SIP-based?	7
4.2.	Why new SIP extensions are needed?	8
5.	Client Protocol Design	9
5.1.	Overview	9
5.2.	Protocols for Accessing Services	9
5.2.1.	Accessing Services Offered by Peers	9
5.2.2.	Accessing Services Offered by Clients	9
5.3.	Event Package Extension	9
5.3.1.	Event packages for clients locating services	10
5.3.2.	Event package for clients announcing services	10
6.	Network Architecture	11
7.	Overview of Operations	12
8.	Event Packages for P2PSIP Clients Locating Services	13
8.1.	Service Type Event Package	14
8.2.	Server Peer List Event Package	14
8.2.1.	Server Peer List	15
8.3.	Persistent Subscription and Immediate Fetch	16
8.4.	Roles in Subscription	16
9.	Operations for P2PSIP Clients Locating Services	16
9.1.	Discover Service Types	16
9.2.	Locate Server Peer	17
9.2.1.	Query Server Peer List	17
9.2.2.	Peers Notify the Changes of Server Peers	17
9.2.3.	Peers Notify Clients Before Leave	18
9.3.	Serving Peer Selection	18
9.3.1.	Collecting Serving Peer Candidates	19
9.3.2.	Selecting Criteria	19
9.3.3.	Peer instructs client to select peer	19
10.	P2PSIP Clients Publish Services	20
10.1.	Service Entry Event Package	21
10.2.	Publish Service Entries	21
11.	IANA Considerations	21
12.	Security Considerations	21
13.	Acknowledgements	22
14.	References	22
14.1.	Normative References	22
14.2.	Informative References	22
	Authors' Addresses	23
	Intellectual Property and Copyright Statements	24

1. Introduction

There are two kinds of nodes in P2PSIP, which are P2PSIP Peer and P2PSIP Client [[I-D.ietf-p2psip-concepts](#)]. However, for a long period of time, the concept of P2PSIP Client and its role have been a controversial source in the discussion of IETF P2PSIP WG.

According to [[I-D.ietf-p2psip-concepts](#)], P2PSIP client is defined as a node participating in a P2PSIP Overlay that is less capable than a P2PSIP Peer in some way. Additionally, P2PSIP Client Protocol is defined as the protocol spoken between P2PSIP Clients and P2PSIP Peers. Besides the definitions, there are quite a lot pros and cons on whether the P2PSIP Client and P2PSIP Client Protocol are necessary. However, despite all these disputes, it has been agreed that P2PSIP Clients do have the ability to add, modify, inspect, and delete information in the overlay if they exist.

In this document, based on the analysis of the heterogeneity among all the nodes which want to use P2PSIP services, we argue that the role of P2PSIP Client is indispensable. P2PSIP clients are suitable role for several categories of nodes, such as the nodes whose P2P module is incompatible with the existing P2PSIP overlay, the nodes which want to provide service but not willing to join in the P2P overlay, the nodes with limited computational resource, the existing SIP devices and SIP-based soft phones, etc. [Section 3](#) explains these different cases.

To make these clients capable of accessing services in P2PSIP overlay, a P2PSIP Client Protocol is needed. We come up with two basic guidelines for our design. Firstly, the proposed client protocol should be SIP-based. Secondly, new SIP extensions are needed to address the new challenges raised by P2PSIP. The reasons for such decisions are presented in [Section 4](#).

Following these guidelines, a client protocol is proposed. This client protocol is mainly based on SIP but with new SIP Event Package Extensions. The high level description on the basic operation of the proposed client protocol, the contents and operations of newly-defined SIP Event Package are also presented. Such an approach not only makes the proposed client protocol compatible with existing SIP devices but also satisfies new challenges raised by P2PSIP.

The detailed design and implementation of this client protocol are still in progress. This version of draft aims at demonstrating the necessity of P2PSIP Client, exploring some new challenges for client protocol raised by P2PSIP, defining basic concepts and mechanisms used in client protocol. In the future visions, more detailed work will be presented.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

2. Concepts and Terminologies

This section describes several key concepts used in this document. Generally speaking, the concepts and terminologies used in this document follow the definitions in [[I-D.ietf-p2psip-concepts](#)]. However, some definitions are modified and some new concepts are added. Both modified and newly-added concepts are listed as follow.

- o P2PSIP Client: a kind of nodes which want to use the services provided by P2PSIP overlay and MAY provide services to other nodes but can not join in P2PSIP overlay due to some reasons. Such nodes MAY be the node which supports P2PSIP but its P2P module is incompatible with the existing P2PSIP overlay, the node that is not qualified to join P2P overlay due to its limited resource, etc. The P2PSIP Client defined in this document is designed to be complied with the general agreement in [[I-D.ietf-p2psip-concepts](#)], that is to say, it do have the ability to add, modify, inspect, and delete information in the overlay.
- o P2PSIP Client Protocol: The protocol spoken between P2PSIP Clients and P2PSIP Peers. It is used to provide a mechanism for exchanging information between clients and peers. Clients can use this protocol to ask their serving peers to add, modify, inspect, and delete information in the overlay on behalf of client itself. Peers can use this protocol to tell clients the information on itself, other peers, as well as the services which have been registered into P2PSIP overlay.
- o Server Peer: a P2PSIP Peer which is capable of providing one or more services to clients.
- o SIP Server Peer: Server Peer that behaves as registrar and SIP proxy. SIP server peer provides SIP service to clients.
- o SIP Service: Service provided by SIP server Peer. This service allows clients to create, modify, terminate sessions with other nodes.
- o STUN Server Peer: Server Peer that behaves as STUN server. STUN server peer provides STUN service to clients.

- o **Serving Peer:** a P2PSIP Peer which is offering certain types of services to client. For example, when a P2PSIP client is using a P2PSIP peer as its SIP outbound proxy, the peer is a SIP serving peer.
- o **Associated Peer & Associated Client:** The server peer(s) on which a P2PSIP client makes persistent subscription(s) of server peer list(s) are referred to as "associated peer(s)" of the client; the P2PSIP client is referred to as "associated client" of the associated peer(s).

3. The necessity of P2PSIP Client

The necessity of P2PSIP Client originates from the heterogeneity among all the nodes which want or need to use P2PSIP services. Generally speaking, these nodes can be divided in to several categories which are discussed in details as follow.

Firstly, the heterogeneity of P2P overlays makes P2PSIP Client necessary. For example, a P2PSIP node which only supports Chord algorithm, needs to access services of P2PSIP overlay which uses Bamboo as P2P algorithm. Due to the incompatible P2P algorithms, the Chord-based node can not join in the Bamboo-based DHT overlay. Thus the Chord-based node has to work as P2PSIP Client.

The second kind of Client candidates comes from the nodes which want to provide services. For example, a conferences server wants to provide conference services to nodes in a P2PSIP system. Because the decoding, mixture and encoding of multiple media streams can be quite a heavy load for its computational resources, this conference server may choose not to join in the P2PSIP overlay. Thus, some extra resources, which would be used for the maintenance of P2P overlay if the server joined overlay, can be utilized for providing better conference services.

Thirdly, the heterogeneity among P2PSIP nodes also makes P2PSIP Client necessary. In order to make P2PSIP overlay convergent and efficient, some kinds of nodes, such as fast-moving nodes, are not suitable to join the overlay. These unqualified nodes can only act as client, using services provided by P2PSIP overlay rather than joining in the overlay.

The fourth kind of P2PSIP Clients come from the vast existing SIP clients, including hardware-based SIP devices and soft phones which support SIP. Obviously, these non-P2P aware SIP clients can neither join in P2PSIP overlay nor offer services. However, for smooth evolution of P2PSIP, these SIP clients must be taken into

consideration at the beginning of P2PSIP's commercially deployment. Therefore, the backward-compatibility with traditional SIP [[RFC3261](#)] is indispensable. P2PSIP Client may be the only suitable role for these non-P2P aware SIP clients.

The cases listed in this section are neither comprehensive nor exclusive; in other words, client may be needed in other scenarios. However, the comprehensive scenarios where P2PSIP Clients are necessary are out of the scope of this document.

4. Our Design Guidelines

In this section, two fundamental guidelines that we follow during the design process of our proposed client protocol are presented. One states that the proposed protocol should be SIP-based; the other one is that new SIP extensions are needed. Each principle addresses one different perspective of challenges which client protocol has to face.

4.1. Why SIP-based?

The first one of our guideline states that SIP is the base of newly-proposed client protocol. There are mainly two reasons for such decision.

Firstly, SIP is needed because the client protocol has to meet the basic session control requirement. The client protocol must support peers registering clients' information into P2PSIP overlay and the establishment, modification and termination of sessions between client and another node within P2PSIP system. In order to satisfy such requirements, SIP is ideal due to its simplicity, flexibility, etc. In addition, SIP has been not only standardized but also widely used; therefore, using the existing well-developed SIP technologies will reduce the complexity for developers implementing the client protocol.

Secondly, in order to make the proposed client protocol compatible with existing SIP devices, SIP is also indispensable. Vast SIP devices and SIP-based soft phones have already been used widely. Thus, to enlarge the user quantity quickly when P2PSIP is deployed at its initial stage, the backward-compatibility which ensures non-P2P aware SIP devices can access services provided by P2PSIP overlay is important.

4.2. Why new SIP extensions are needed?

Although the basic session control functions can be implemented by existing SIP, there are still lots of other challenges raised by new scenarios of P2PSIP. The challenges which have been identified are explained as follow.

- o The service reliability requirement. In [RFC 3261](#) [[RFC3261](#)], the outbound proxy is typically manually configured on SIP UA. However, P2PSIP peers are typically unstable and ephemeral compared with SIP proxies in traditional SIP; they may leave the overlay or crash from time to time. Therefore, to ensure that clients can access services reliably even though the peers offering this service are unreliable, new SIP extensions are needed. For example, client can use SIP Event Notification Mechanism [[RFC3265](#)] to get all available server peers. If the one currently used crashed suddenly, client can switch to another available one.
- o The service availability requirement. As mentioned in [[I-D.ietf-p2psip-concepts](#)], some individual peers may provide extra services, for example STUN [[I-D.ietf-behave-rfc3489bis](#)] services, to the overlay or members in the overlay. To make these services accessible for clients, the client protocol MUST enable peers to tell clients what services are available and the detailed parameters for some services of client's interest.
- o The service efficiency requirement. Both client and peer need to operate effectively. Taking the load of peers for example, client protocol MUST enable peers indicating clients to choose other peers based on some criteria when the peer becomes overloaded. Similarly, in order to ensure the client's efficiency, the client protocol MUST allow client to choose the most suitable peer as its serving peer. The term "most suitable" can be interpreted as low RTT, high bandwidth, etc. However, the exact meaning of the term "most suitable" SHOULD be determined by the local policy of client which is out of the scope of this document
- o The service announcing requirements. The client protocol MAY be able to be used for announcing the services that clients offer to the P2PSIP overlay or members of the overlay. It is still an open issue that whether P2PSIP clients should provide services or not, and this issue needs further study. However, if P2PSIP clients provide services, service announcing mechanism needs to be defined in client protocol to support these services. What service P2PSIP clients may offer is still an open issue. Storage is a possible service proposed in [[I-D.ietf-p2psip-concepts](#)]. Clients may also help in bootstrap procedure of other peers and/or clients.

All these above requirements cannot be satisfied by existing SIP and its extensions, thus new SIP extensions are needed, which become our second guideline in designing a client protocol.

5. Client Protocol Design

5.1. Overview

The client protocol proposed in this document is extended from SIP protocol. Currently we define three new SIP event packages. Additional extensions may need to be added in client protocol in the future.

5.2. Protocols for Accessing Services

5.2.1. Accessing Services Offered by Peers

P2PSIP overlay provides services for P2PSIP clients. P2PSIP clients access the services through P2PSIP server peers, and P2PSIP server peers behave as servers, such as SIP servers and STUN servers.

Conventional SIP and related protocols allow P2PSIP clients to use the basic services provided by overlay. For example, SIP enables clients to create, modify and terminate sessions with other nodes in overlay. ICE [[I-D.ietf-mmusic-ice](#)] and STUN enable sessions to traverse NATs.

P2PSIP overlay MAY provide some additional services to P2PSIP clients. To allow P2PSIP clients accessing these services, other protocols might be used between peers and clients or some extensions might be added to client protocol.

5.2.2. Accessing Services Offered by Clients

P2PSIP clients MAY provide services. To allow other nodes accessing these services, other protocols might be used between peers and clients or some extensions might be added to client protocol.

5.3. Event Package Extension

We define three SIP event packages to meet the requirements in P2PSIP described in 4.2. P2PSIP clients can utilize these event packages to locate services and announce services. These SIP event packages are extendable, that is to say, these extensions don't require any modification of SIP protocol implementations. Thus, they are easy to implement.

[5.3.1.](#) Event packages for clients locating services

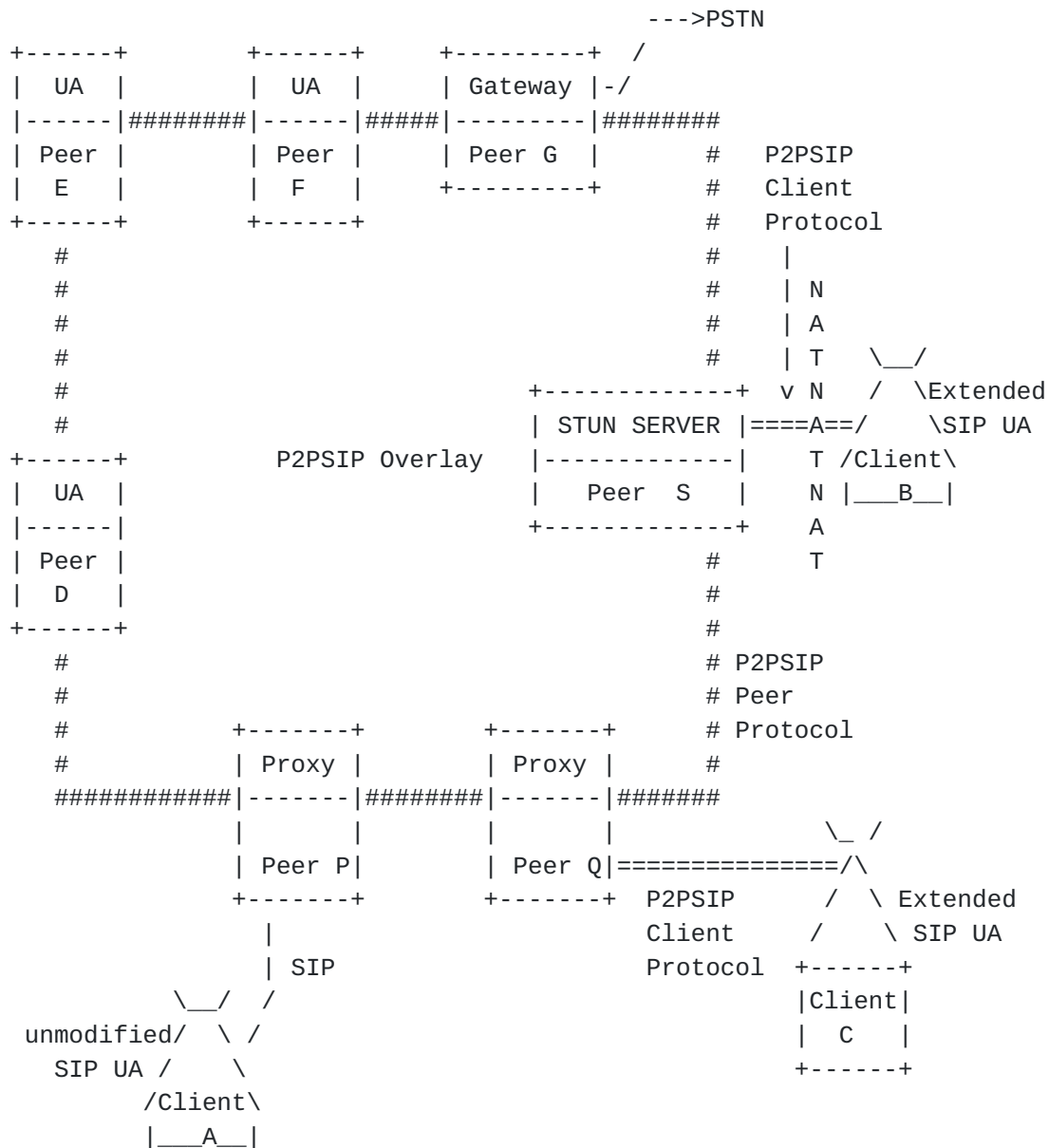
Two event packages are defined in [section 8](#) for clients to locate service. Operations based on these two event packages are described in [section 9](#). These operations meet the service availability, availability and efficiency requirements described in 4.2. These operations are listed below:

- o Operations required by service availability. P2PSIP clients discover what services the overlay provides. P2PSIP clients locate server peers that can provide specific service.
- o Operations required by service reliability. P2PSIP peers notify clients the changes of server peers. P2PSIP peers notify clients before leaving.
- o Operations required by service efficiency. P2PSIP clients select serving peers. P2PSIP clients search desirable server peers. P2PSIP peers instruct clients to select serving peers.

[5.3.2.](#) Event package for clients announcing services

An event package is defined in [section 10](#) to meet the services announcement requirement described in 4.2. Clients can announce services based on this event packages. The details are described in [section 10](#).

6. Network Architecture



P2PSIP Network Reference Model

Figure 1

Fig 1 illustrates the architecture of P2PSIP networks. This figure is mainly based on the "Figure: P2PSIP Overlay Reference Model" in [I-D.ietf-p2psip-concepts]. In addition, some modifications are made in order to highlight different types of P2PSIP clients and the

P2PSIP Client Protocol.

In Fig 1, each node is separated into two parts. The upper part describes the functionality of this node; while the lower part tells the node's role, client or peer.

Several peers construct a P2PSIP overlay defined in [[I-D.ietf-p2psip-concepts](#)] using peer protocol. Some of these peers are coupled with SIP proxy, thus they can be used as SIP outbound proxies by clients. Some other peer may provide extra services, such as STUN/TURN services.

Client does not join the P2PSIP overlay but use the services provided by P2PSIP overlay. Three types of clients are illustrated in Fig 1, which are unmodified SIP UA, extended SIP UA and extended SIP UA located behind NAT devices.

Unmodified SIP UA refers to the SIP UA which do not support P2PSIP Client Protocol defined in this document. For such client, denoted as "Client A" in Fig 1, it treats "peer P", which is a peer coupled with SIP Proxy, as a SIP outbound proxy and communicate with other nodes using SIP.

Extended SIP UA refers to SIP UA with extensions defined in this document to support P2PSIP client protocol. For such clients, denoted as "Client C" in Fig 1, they can get information about available alternate peers and other available services in addition to the services unmodified SIP UA can get from "peer Q". The "peer Q" is a peer support both peer protocol and client protocol.

Extended SIP UA may be located behind NAT devices, for example "Client B" in Fig 1. In this case, the client itself has to be STUN/TURN-enabled. When it starts, the client could obtain the address of "peer S", which provides STUN/TURN services. Afterwards, the client can use ICE for establishing sessions based on STUN/TURN services.

7. Overview of Operations

This section represents the basic operations of the proposed client protocol. Generally speaking, these operations can be divided into two categories. The first category contains operations for accessing services, which contains the operations defined in [[RFC3261](#)]. The other category contains operations related to locating services.

The first kind of operations includes,

- o SIP service operations. The client acts as SIP UA, and its serving peer acts as registrar and SIP proxy. These operations include client registering itself to peer, peer proxying client's requests, etc. These operations are defined in SIP [[RFC3261](#)] and some existing extensions.
- o Operations for accessing other services offered by peers. These operations need further study.
- o Operations for accessing the services offered by clients. These operations need further study.

The second kind of operations includes but not limited to,

- o Client discovers available service types provided by P2PSIP overlay
- o Client queries the list of available server peers
- o Peers inform clients the changes of server peers
- o Peers inform clients before leaving P2PSIP overlay
- o Clients select serving peers
- o Peers instruct clients to select serving peers
- o Clients announce services

This kind of operations needs new SIP extensions. In this document, we propose extending SIP Event Package [RFC 3265](#) [[RFC3265](#)] to achieve this goal. The high level definition of new event packages and their operations are described in [section 8,9,10](#).

8. Event Packages for P2PSIP Clients Locating Services

As discussed above, existing protocols can not meet all the requirements of client protocol. To meet the services availability, reliability and efficiency requirements described in 4.2, two event packages are defined in this section. Operations based on these event packages are described in next section. Every server peer MUST support these extensions. But supporting these SIP extensions doesn't mean that a server peer MUST work as registrar and proxy.

In this document, event packages are utilized to transfer service information and server peer information from P2PSIP peers to P2PSIP clients. A new event package called STEP (Service Type Event

Package) is defined for transferring information about services types from P2PSIP peers to P2PSIP clients. A new event packages called SPLEP (Server Peer List Event Package) is defined for transferring information of server peers from P2PSIP peers to P2PSIP clients. This document only gives high level descriptions on the information contained in STEP and SPLEP. Separated documents SHOULD specify the formats of STEP and SPLEP in details.

8.1. Service Type Event Package

A new event package called STEP (Service Type Event Package) is defined for transferring service type information from P2PSIP peers to P2PSIP clients. A STEP lists all service types that a P2PSIP overlay can provide to P2PSIP clients. A STEP also contains the overlay's overlay name. We give an example to show the information contained in a STEP as bellow:

Overlay Name: example.chord
Service Type List: SIP, STUN

To generate STEP, peers SHOULD know all service types the overlay provides to P2PSIP clients. P2PSIP peer protocol SHOULD provide a mechanism to make peers aware of these services. This issue is out of the scope of this document.

8.2. Server Peer List Event Package

A new event packages called SPLEP (Server Peer List Event Package) is defined for transferring information of server peers from P2PSIP peers to P2PSIP clients. SPLEP contains overlay name and one or more server peer list. For example:

Overlay Name: example.chord
Service Peer List(s): SIP server peer list, STUN server peer list

Different server peer list in a SPLEP contains information of different type of server peers. Server peer list can be cataloged by the type of server peer. For example, SIP server peer list contains information of SIP server peers, and STUN server peer list contains information of STUN server peers. As a server peer MAY provide more than one type of service, a server peer's MAY be included in different types of server peer lists. By setting parameters in SUBSCRIBE message's body, a P2PSIP client can specify the type(s) of server peer list(s) that should be contained in SPLEP.

8.2.1. Server Peer List

In server peer list, the information of a server peer at least includes server peer type, service locating point, service entry (or service entries). The information of a server peer MAY include two optional attributes, weight and priority, and some service-specific information. An example of SIP server peer list is given below:

Server Peer Type	Service Locating Point	service entries	Priority (optional)	Weight (optional)
SIP	30.3.3.3:5060, UDP;	30.3.3.3:5060, UDP;	1	1001
	30.3.3.3:5060, TCP	30.3.3.3:5060, TCP		
SIP	20.2.2.2:9060, UDP;	20.2.2.2:9060, UDP;	1	1000
	20.2.2.2:9060, TCP	20.2.2.2:9060, TCP		

Table 1: SIP sever peer list example

Explanations of server peer's attributes:

- o Server peer type: Server peer type here is corresponded with the type of server peer list containing the server peer. For example, in a SIP server peer list, every server peer's server peer type is set to SIP.
- o Service entry: A service entry includes IP address, port, transfer protocol. A server peer MAY have several service entries that provide the same type of service. For example, a SIP server peer MAY provide SIP service via TCP and UDP at the same time.
- o Service locating point: Service locating point includes IP address, port, transfer protocol. Service locating point is the entry for P2PSIP clients subscribing STEP and SPLEP. Service locating point MAY or MAY not be the same as SIP service entry.
- o Weight and Priority: Weight and Priority attributes indicate a server peer's capacity and priority separately. The meanings of these attributes are similar with those in DNS SRV records [RFC 2782](#) [RFC2782]. Both the Priority attribute and Weight attribute are numbers. The Priority attribute indicates the priority that a

peer SHOULD be selected. Lower numeric Priority value means higher priority to be selected. The Weight attribute indicates a peer's capacity to accommodate clients. The peer with higher Weight can accommodate more clients. Peers MAY instruct clients to select serving peers by utilizing these two attributes. Priority and Weight attributes MAY be variables, and server selection MAY be dynamic.

8.3. Persistent Subscription and Immediate Fetch

With SIP SUBSCRIBE/NOTIFY methods [[RFC3265](#)] or PUBLISH method [[RFC3903](#)], event packages can be transferred between SIP entities. In this document, SUBSCRIBE/NOTIFY methods are used to transfer STEP and SPLEP. SIP SUBSCRIBE/NOTIFY mechanism provides two types of subscriptions: persistent subscription and immediate fetch.

An immediate fetch is a subscription expires immediately. Immediate fetch can be used to fetch current status of the event. An immediate fetch is effected by sending a SUBSCRIBE with an "Expires" header of 0.

Persistent subscription lasts for some time. If subscribers make persistent subscription, notifiers will notify subscribers every time when the status of the event changes. Both persistent subscription and immediate fetch are used in this client protocol.

8.4. Roles in Subscription

Every P2PSIP server peer MUST have at least one service locating point, on which P2PSIP clients can subscribe STEP and SPLEP. If a P2PSIP client subscribes STEP, SPLEP or both, it behaves as subscriber. A P2PSIP server peer MUST behave as a notifier, if any P2PSIP client subscribes STEP or SPLEP from it.

9. Operations for P2PSIP Clients Locating Services

9.1. Discover Service Types

P2PSIP client can discover available service types by subscribing STEP (Service Types Event Package) from P2PSIP peers. By parsing STEP, a P2PSIP client can find out all service types the overlay can provide. If the P2PSIP client wants to use any type of service among these services, it can use the way described in 8.2 to locate server peers.

9.2. Locate Server Peer

SPLEP contains information of one or more types of server peers. In SPLEP, the information of a server peer at least includes its overlay name, server peer type, service locating point, service entry (or service entries), weight and priority. By setting parameters in SUBSCRIBE message's body, a P2PSIP client can specify the type(s) of server peer list(s) that should be contained in SPLEP. By subscribing SPLEP and parsing it, P2PSIP clients can find server peers of given types and the service entries on these server peers. A P2PSIP client MAY subscribe SPLEP from several peers. A P2PSIP client MAY subscribe several types of server peer lists. A P2PSIP client MAY subscribe the same type of server peer lists from different peers.

For redundancy, any server peer list in a SPLEP SHOULD contain several server peers if possible. To generate SPLEP, P2PSIP peers MUST collect information about available server peers from other peers. But a P2PSIP peer doesn't need to find all available server peers in the overlay, and it's impossible usually. These issues are out of the scope of this document. Some needed mechanisms SHOULD be defined in P2PSIP peer protocol.

Immediate fetch or persistent subscription MAY be used to locate server peer. Immediate fetch and persistent subscription are used to support different operations as described in 9.2.1, 9.2.2 and 9.2.3.

9.2.1. Query Server Peer List

Immediate fetch SHOULD be used for querying server peer list. P2PSIP clients MAY query server peer list occasionally. For example, a P2PSIP client seldom uses some type of service. To reduce overhead, the P2PSIP client may query this type of server peer list every time before it wants to use this type of service.

9.2.2. Peers Notify the Changes of Server Peers

P2PSIP server peers are unstable, transient. They may leave or fail any time. New server peers may join the overlay anytime. If P2PSIP clients want to be notified the changes of server peers, they SHOULD make persistent subscriptions.

Suppose a client make a persistent subscription on a peer. To generate SPLEP for persistent subscription, the peer MUST generate and maintain the server peer list(s) required in SPLEP. The peer MUST has the ability to detect if any server peer in the server peer list(s) leaves or any server peer (exclude itself) in the server peer list(s) fails. The leaving peer or failed peer MUST be removed from

the server peer list(s). The peer MAY add one or more server peers in the server peer list(s) it maintains. When the server peer list(s) changes, the peer SHOULD notify its associated clients.

9.2.3. Peers Notify Clients Before Leave

Persistent subscriptions are required for peer notifying clients before leave. Suppose a P2PSIP client makes persistent subscription of SPLEP from a peer. The client subscribes one or more server peer lists in the persistent subscription. Then the client can receive notification if any server peer in the server peer list(s) is about to leave. We discuss the notification in two cases:

Suppose the leaving peer is not the associated peer. If the associated peer detects a peer in the server peer list(s) is about to leave, the associated peer notify the client a SPLEP from which the leaving peer is removed. Notification in this case requires the associated peer has the ability to detect if any peer in the server peer list(s) is about to leave. The detect means MAY be the leaving peer notifying the associated peer. The detect means is out of the scope of this document.

Suppose the leaving peer is the associated peer. To leave gracefully, the leaving peer SHOULD inform the associated client about its leaving and terminate the subscription. This can be done by sending the associated client a SPLEP with NOTIFY method. The leaving peer is removed from the server peer list(s) in the SPLEP. To terminate subscriptions, the NOTIFY message contains a "Subscription-State" header with a value of "terminated" as described in [RFC 3265](#) [[RFC3265](#)].

9.3. Serving Peer Selection

For some reason, such as getting better services, load balance, a P2PSIP client MAY select one or more server peers from all the reachable server peers it knows as serving peer. And other reachable server peers the P2PSIP client knows are backups of serving peers. P2PSIP clients MAY select serving peers periodically. P2PSIP clients MAY select serving peers every time before requesting services.

A P2PSIP client MAY select one or more types of serving peers, such as SIP serving peer, STUN serving peer. Specific type of serving peer selection is among the same type of server peers. For example, SIP serving peer selection is among SIP server peers. In the rest of this section, if no specification, serving peer selection refers to a specific type of serving peer selection.

9.3.1. Collecting Serving Peer Candidates

For selecting serving peers, a client needs to subscribe SPLEP from one or more peers. If the client subscribes SPLEP from more peers, it MAY find more serving peer candidates. Then, it can find more backup peers and MAY find serving peers which are more desirable. If a P2PSIP client subscribes SPLEP from a peer for collecting serving peer candidates exclusively, immediate fetch SHOULD be used to reduce overhead.

9.3.2. Selecting Criteria

Selecting serving peers can be based on any criteria, such as latency, SIP session success rate, instruction from server peer. The selecting criterion MAY be dynamic and MAY vary in different P2PSIP overlays. Establishing such a criterion is out of the scope of this document.

9.3.3. Peer instructs client to select peer

Peers MAY or MAY not give instructions to P2PSIP clients for selecting peers. This document provides the mechanism that peers instruct clients to select peers. This mechanism can be used to distribute the requests initiated by clients among server peers. This mechanism MAY help to avoid the situation that some peers are overload, to achieve load balancing. It also MAY help clients to get better services.

As avoiding overload and load balance involve many factors, other mechanisms are also needed. Except for serving clients, P2PSIP peers have other tasks such as maintaining overlay and storing data. The capacities of P2PSIP peers are also heterogeneous. Some mechanisms MAY be deployed in peer protocol for avoiding overload and load balancing. These issues are out of the scope of this document.

Peers can instruct P2PSIP clients to select serving peers by utilizing Priority and Weight attributes. There are two types of instruction: strict instruction and loose instruction.

o Strict instruction

In strict instruction case, the criterion of selecting peers is decided by peers. P2PSIP clients receive instruction from peers and select peers strictly obeying instructions if possible. The rules of strict instruction are the same as the rules in selecting servers defined in [RFC 2782](#) [[RFC2782](#)]. Among all the reachable peers, P2PSIP clients MUST select serving peer from the peer(s) with lowest Priority attribute (i.e. highest priority). If there are more than

one peers with highest priority, P2PSIP clients SHOULD select according to these peers' weight attributes. Larger weights SHOULD be given a proportionately higher probability of being selected. An example is given as follow. All reachable SIP server peers a P2PSIP client knows is listed as below:

Service Peer	Priority	Weight
SIP server peer A	1	30
SIP server peer B	1	20
SIP server peer C	2	100

Table 2: strict instruction example

As server peer A and server peer B have highest priority (lowest Priority attribute value), the serving peer must be selected from them. According to their Weight attributes, the probability of server peer A being selected should be $30/(30+20)$, and the probability of server peer B being selected should be $20/(30+20)$.

From the view of peers, P2PSIP peers MAY set priority and weight attributes with the consideration of load distribution. As P2PSIP clients select peers only according to Priority and Weight attributes, P2PSIP peers MAY set Priority and Weight attributes with other considerations so that P2PSIP clients can get better services. For example, proximity-aware technologies MAY be deployed and P2PSIP clients can be instructed to select close peers.

o Loose instruction

In loose instruction, P2PSIP clients select peers with the considerations of other factors besides priority and weight attributes. The selection criterion is out of the scope of this document. But peers with higher priority and peers with higher capacity SHOULD have more chances to be selected.

10. P2PSIP Clients Publish Services

P2PSIP clients MAY or MAY not provide some services, such as storage. This document is not intent to discuss what services P2PSIP clients SHOULD provide and how to implement these services. These services may be accessed through existing protocols or client protocol. These issues are out of the scope of this document. We only describe how to publish the services provided by P2PSIP clients, if these services exist. In this section, a new SIP event package called SEEP (Service

Entry Event Package) is defined for transferring services information.

10.1. Service Entry Event Package

A SEEP contains information of all entries of services provided by a P2PSIP client. SEEP also contains some related information, such as overlay name. In SEEP, information of a service at least includes service type, service entry (or service entries). Service-specific information and service capacity information MAY be contained in SEEP. Suppose a P2PSIP client provides storage service via FTP, the SEEP MAY look like:

Service Type	Service Entry	Service-specific Information
FTP	10.0.0.2:21,TCP	username: alice; password: PasswordOfAlice

Table 3: SEEP example

10.2. Publish Service Entries

P2PSIP clients MAY send SEEP to P2PSIP peers with SIP PUBLISH method. P2PSIP peers MAY or MAY not publish the service information of P2PSIP clients in overlay. P2PSIP clients MAY access the services provided by other clients just like access the services provided by peers.

11. IANA Considerations

This document proposes several event packages, each of them need IANA considerations such as Package Name, Package or Template-Package, etc. However, because this document only provide a high level description of these event packages, the detailed description of each event package and its own IANA considerations will be defined in respective documents. Thus, no IANA consideration is listed in this document.

12. Security Considerations

Although some attack and threat models and security mechanisms for SIP are proposed in [RFC 3261](#) [[RFC3261](#)], there are still lots of security challenges which the proposed P2PSIP Client Protocol has to face. What is more, compared with Client/Server mode, the nature of

peer-to-peer system makes it even more complicated to design a secure client protocol.

Currently, we are at the initial stage of designing security mechanism for the proposed client protocol. Thus, in this version of draft the security issues are not covered. In the future version, the security considerations will be presented in a detailed way.

13. Acknowledgements

The authors wish to thank Tao Ma and Zhenyu Wu for their valuable comments. In addition, the authors also quite appreciate the discussions on the topic of P2PSIP Client in the IETF P2PSIP maillist.

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", [RFC 2782](#), February 2000.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.
- [RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [RFC3903] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", [RFC 3903](#), October 2004.

14.2. Informative References

- [I-D.ietf-behave-rfc3489bis]
Rosenberg, J., Huitema, C., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for (NAT) (STUN)", [draft-ietf-behave-rfc3489bis-11](#) (work in progress), October 2007.

- [I-D.ietf-mmusic-ice]

Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols",
[draft-ietf-mmusic-ice-19](#) (work in progress), October 2007.

[I-D.ietf-p2psip-concepts]

Bryan, D., "Concepts and Terminology for Peer to Peer SIP", [draft-ietf-p2psip-concepts-00](#) (work in progress), July 2007.

Authors' Addresses

LiChun Li
Mobile lIfe and New mEdia Lab, BUPT.
P.O. Box 92, No.10, Xitucheng Road
BeiJing, Haidian District 100876
China

Email: lilichun@gmail.com

Chunhong Zhang
Mobile lIfe and New mEdia Lab, BUPT.
P.O. Box 92, No.10, Xitucheng Road
BeiJing, Haidian District 100876
China

Email: zhangch@bupt.edu.cn

Yao Wang
Mobile lIfe and New mEdia Lab, BUPT.
P.O. Box 92, No.10, Xitucheng Road
BeiJing, Haidian District 100876
China

Email: yaowang.bupt@gmail.com

Yang Ji
Mobile lIfe and New mEdia Lab, BUPT.
P.O. Box 92, No.10, Xitucheng Road
BeiJing, Haidian District 100876
China

Email: jiyang@bupt.edu.cn

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

